**Black Box Competition: Documenting the Journey and Insights**

**Alfonso Camacho Bustillo – Id 544**

Participating in the black box competition over the last three months has been quite the adventure. It involved a process of trial, error, and iteration to understand the dynamics of eight different functions and figure out strategies to maximize each function’s outputs. Here’s a breakdown of my journey, covering the initial steps, the learning curve, the challenges, and the victories along the way. I'll also share some of the code and tools I used, which are attached for reference.

**Initial Steps**

**Approach and Initial Code from Classes**

When the competition started, I quickly realized that having a structured plan was crucial because we only had a limited number of iterations and feedback opportunities. My initial understanding was mostly based on intuition and high-level concepts, rather than deep technical knowledge. As suggested by our facilitators, I explored various academic papers and external code sources. However, most of these were too technical for my current understanding, so I decided to start with the basic models and libraries introduced in class. This approach allowed me to build on a familiar foundation and gradually incorporate improvements.

**Simulator for Strategy Testing** *(Simulators attached with all files)*

Using the basic models from class, I developed a simulator to replicate the exercise. This simulator allowed me to create different functions and test various strategies by adjusting Kernel parameters and Acquisition function variations. It was a great way to see how different strategies might play out over 25 iterations under various scenarios. The attached simulator codes were instrumental in helping me understand the impact of different kernel parameters, particularly length scale and noise assumptions, on the model’s performance. This understanding formed the basis for my initial models and subsequent improvements.

**Analytical Exploration**

To ensure a deep understanding of each function's dynamics, I implemented comprehensive analytics to explore the data. This step was crucial throughout the competition as it helped identify specific function characteristics, necessary data transformations, and potential areas of concentration. The analytics also revealed if certain regions were being over-explored or under-explored, guiding my strategy adjustments. Initially, this exploration helped me see if I was concentrating too much in certain areas and not enough in others. Also, examining the correlation between inputs and outputs, and occasionally looking at quadratic correlations, helped uncover less obvious dependencies.

**The First Model**

**Source Code and Procedure** *iInitial codes attached with files)*

My initial code, based on class lessons, followed a straightforward procedure:

1. Create a grid for exploration.
2. Define a Kernel (RBF) with specific length scale and noise assumptions.
3. Fit a Gaussian Process Regressor with the defined Kernel.
4. Generate a surrogate model over the grid.
5. Apply Acquisition functions: Upper Confidence Bound (UCB), Expected Improvement (EI), and Probability of Improvement (PI).

**Challenges and Evolution**

The primary challenge was determining the Kernel parameters. I started doing this manually, based on course suggestions and insights from my simulator. This method, however, proved to be volatile and inconsistent. Additionally, there were numerous Kernel variations and parameterizations that I didn’t fully understand, requiring a trial-and-error approach. For the acquisition functions, UCB showed more predictable behavior in my simulations, so I focused on it with broad exploration parameters, while still running EI and PI to gather additional information.

**Specific Function Learnings and Adaptations** *(Code per function attached with files)*

Each function presented unique challenges and required specific adaptations:

1. **Function 1**: This function involved a bit of luck as areas with contamination were closely grouped. I balanced exploration and exploitation from the start and experimented with various data transformations to highlight regions with non-zero outputs. Ultimately, a combination of regular scaling and additional data helped improve my results.
2. **Function 2**: After noticing significant noise in the outputs, I started averaging historical outputs for each observation. Later, integrating all historical data into the database significantly improved the model’s robustness. Towards the end, I tested two new models: one with all historical observations and another using grouped inputs with mean and standard deviation. Integrating all historical data proved most effective.
3. **Function 3**: Similar to Function 2, managing random outputs through historical data integration was key.
4. **Function 4**: Early exploration revealed a good maximum, but finding a better output was challenging. I noticed that positive and negative outputs were on different scales, so I adjusted the scaling, which improved performance.
5. **Function 5**: Initial exploration quickly revealed the maximum at the edge of input ranges, simplifying subsequent queries.
6. **Function 6**: Similar noise management strategies as Functions 2 and 3.
7. **Functions 7 and 8**: High dimensionality required consistent refinement through regional exploration and feature reduction. Initial effectiveness waned over time but remained useful for monitoring relevant areas.

For Functions 2, 3, and 6, I provided two versions of the code: one with the initial average methodology and another with all historical feedback integrated and modeled. Each function's code includes a record of every submission and all candidate queries.

**Evolution to the Most Recent Models**

Around week six, I made a significant breakthrough by optimizing the log-likelihood of fitting the Gaussian Process Regressor. This involved an RBF + White Kernel structure to evaluate different levels of noise and length scale, which provided an optimal setup for each function. Transitioning to the log-likelihood version took several iterations, but it significantly enhanced the model’s efficiency.

In detail, this process involved evaluating the log-likelihood for various combinations of kernel parameters, selecting those that maximized the fit quality. This approach provided a more systematic and reliable way to determine the best kernel parameters, compared to the earlier manual and trial-and-error methods. As a result, my models became more robust and predictive, improving the accuracy of the surrogate models.

**Final Results and Reflections**

**Performance and Lessons Learned**

In the final weeks, my scores improved notably due to strategic refinements and the implementation of the log-likelihood optimization. If I had more weeks, I would have automated the decision process for query selection, enhancing efficiency. Starting over, I would focus on better noise management and adaptability to specific function characteristics, potentially incorporating heteroscedasticity modeling and automated data transformation trials.

**What Worked Well**

* **The Log-Likelihood Model for Kernel Optimization**: This was the best decision I made, significantly improving performance and providing a more systematic way to determine optimal kernel parameters.
* **Submitting Two Queries Per Week**: Instead of waiting until the end, submitting two queries per week allowed for broader exploration and model calibration. This strategy enabled me to test different model setups simultaneously and gather more data to refine my approach. As the competition progressed, I reduced to one submission per feedback session, which was extremely useful for gathering data from the noisier functions.
* **Regularly Reviewing Analytics**: Post-feedback analytics ensured that my strategy remained aligned with function dynamics, helping identify areas for improvement and guiding future adjustments.
* **Upper Confidence Bound Acquisition Function**: UCB was consistent and controllable, allowing me to balance exploration and exploitation effectively.

**Challenges and Areas for Improvement**

* **EI and PI Acquisition Functions**: These functions were inconsistent in the queries they suggested. While EI occasionally provided unexpected good outputs, I struggled to integrate it effectively into my strategy. Understanding how to leverage EI and PI better remains a future learning opportunity. *iInitial codes attached with files)*
* **BoTorch Implementation**: Despite multiple attempts, integrating BoTorch faced compatibility and numerical issues. Although I had to abandon it, the trials (attached) provided valuable insights into potential improvements. *(Some trials attached with files)*

**Potential Improvements**

If the competition continued, I would automate the decision-making process for query selection, reducing manual input and enhancing efficiency. Starting over, I would focus on:

* **Noise Management**: Implementing better noise handling techniques and adaptability to specific function characteristics.
* **Automated Data Transformation**: Integrating automated trials for data transformations to identify necessary adjustments more systematically.
* **Enhanced Kernel Optimization**: Building on the log-likelihood approach, exploring more complex kernel structures and combinations to further improve model accuracy.

**Conclusion & Final thoughts**

Participating in this black box competition has been an enriching journey of learning and strategic growth. With multiple top-tier finishes and a deepened understanding of Gaussian processes and Kernel optimization, I am well-prepared for future challenges. The insights gained will undoubtedly enhance my approach to machine learning and AI models, fostering continued growth and success.

This competition wasn’t just about scoring high or finding the maximum of the functions. It was a deep dive into understanding the mechanics of machine learning models, the nuances of Gaussian processes, and the strategic thinking required to balance exploration and exploitation effectively. I’ve come out of this experience with a toolkit of strategies and a mindset ready to tackle more complex problems, whether in competitions or real-world applications. The journey was challenging, but the knowledge and skills gained are invaluable.